SELF-ORGANIZING DOUBLY LINKED LISTS[1]

by

D. Matthews[2], D. Rotem[3]

and

E. Bretholz[3]

Research Report CS-78-53

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

December 1978

1 Work carried out under Natural Sciences and Engineering Council of Canada Grants Nos. A-3590 and A-3055.

2 Department of Statistics
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

3 Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

Abstract:

The problem of determining a heuristic which maintains a doubly linked list in approximately optimal order with respect to mean search time is considered. Within a general framework of simple assumptions it is shown that in one particular case no such heuristic can be found, while in many other situations heuristics for similarly maintaining sequential lists should be used. In the remaining circumstances a heuristic known as move to end is shown to reduce search time, on average, to at most twice the minimum value determined by an optimal ordering of the list.

Keywords: Doubly linked lists, searching, self-organizing heuristics, move to end.

CR Categories: 3.74, 4.43, 5.25

In recent years considerable interest has focussed on heuristics for maintaining a sequential list in approximately optimal order. In particular, the 'move to front' and 'transposition' heuristics have been intensively analyzed by Hendricks [4], Burville and Kingman [2], Bittner [1], Knuth [6] and Rivest [7], to name only a few. Present experimental evidence suggests that the transposition heuristic is optimal, and the paper of Rivest succinctly summarizes existing results on this important problem.

One direction in which this same problem can logically be extended is to the data structure known as a doubly linked list. As defined by Knuth [5] a doubly linked list is a linear list in which each node contains two links which point to items in the list on either side of that node (see Figure 1). Such lists often include a list head which simplifies manipulation of the list, but for our purposes it is convenient to suppose that a doubly linked list has well-defined left and right end nodes.

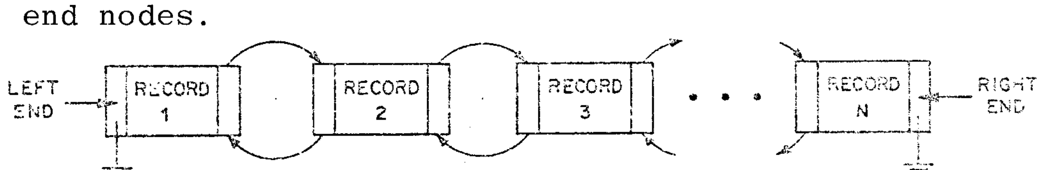

Figure 1.

Knuth [5] remarks that the origin of doubly linked lists is obscure. However, this obscurity is no doubt attributable to their usefulness as an information structure. (For algorithms and possible applications see also Tremblay and

Sorenson [8]). The network model described in the Data Base Task Group Report [3] makes extensive use of doubly linked lists. For this reason the results which follow may prove both academically interesting and practically significant.

A given doubly linked list, D, consists of N records $R_1, R_2, \ldots, R_N$, with p_j the probability of requesting R_j, $j = 1, \ldots, N$. Successive requests for records in D are assumed to be independent and identically distributed, and without loss of generality we suppose that $p_1 \geq p_2 \geq \ldots \geq p_N \geq 0$. Since D is doubly-linked a search for R_i can begin from the leftmost record in D or its rightmost counterpart; let $p_{iL}, p_{iR} = 1 - p_{iL}$, $1 \leq i \leq N$ be the probabilities of these two events, respectively. We assume that the starting point for any search is selected independently of the request for the record being sought.

Suppose, initially, that the probabilities p_i, p_{iL} (and therefore p_{iR}) are known. The following lemma characterizes the optimal ordering of the records if the mean search time, E(S), is to be minimized.

Lemma 1. E(S) is minimized when the records in D are arranged, from left to right, in decreasing order of $p_i^* = p_i(p_{iL} - \frac{1}{2})$.

Proof: Let A be a given arrangement of $R_1, \ldots, R_N$ and let

ℓ_i be the number of records to the left of R_i in A. Then the expected search time for A is

$$E_A(S) = \sum_{i=1}^{N} p_i p_{iL}(\ell_i + 1) + \sum_{i=1}^{N} p_i p_{iR}(N - \ell_i)$$

$$= 2\sum_{i=1}^{N} p_i(p_{iL} - \tfrac{1}{2})\ell_i + N - (N-1)\sum_{i=1}^{N} p_i p_{iL} \qquad (1.1)$$

and (1.1) is minimized when $\sum_{i=1}^{N} p_i(p_{iL} - \frac{1}{2})\ell_i$ is least i.e. when the records are arranged, from left to right, in decreasing order of $p_i^* = p_i(p_{iL} - \frac{1}{2})$.

Two further observations concerning Lemma 1 can also be made. If $p_{iL} = p$ for all i then the optimal arrangement is $R_1, R_2, \ldots, R_N$ as in the case of the sequential list. In particular, if $p_{iL} = \frac{1}{2}$ for all i, $E_A(S) = \frac{1}{2}(N+1)$ does not depend on $\{\ell_i\}$ indicating that, with respect to mean search time, the ordering of the records is immaterial if either end of D is equally likely to be the starting point of a search for any particular record.

In most circumstances the probabilities p_i, p_{iL} $1 \le i \le N$ are unknown and so the records cannot be arranged in the optimal order in advance. Therefore the problem becomes one of finding a heuristic which dynamically maintains D in approximately optimal order. Subsequent developments will show that two different situations arise, depending on the values of the probabilities p_{iL}. In order to discuss and compare such heuristics the following definitions are required.

Definitions.

(a) A *heuristic* for a given doubly linked list, D, is a set of permutations $\tau = \{\tau_j^L, \tau_j^R\}$ such that $\tau_j^L(\tau_j^R)$ is applied to D whenever the requested record is found in position j by a search beginning from the left (right) end of D.

(b) A heuristic $\tau = \{\tau_j^L, \tau_j^R\}$ is said to be *optimal* iff $E_\tau(S)$, the asymptotic expected search time using τ, satisfies $E_\tau(S) \le E_\sigma(S)$ for all probability frameworks $\{p_j, p_{jL}\}$ and all heuristics σ which operate on D.

The first of the two above-mentioned situations concerns a characterization of the optimal heuristic when $p_{iL} = p$ for all i; without loss of generality we take $p \ge \frac{1}{2}$.

Lemma 2.

(i) If $p_{iL} = p = \frac{1}{2}$ no heuristic is necessary to maintain D in optimal order.

(ii) If $p_{iL} = p > \frac{1}{2}$ and if τ is an optimal heuristic for the corresponding sequential list of N records ($p = 1$), then τ is optimal for D as well.

Proof: We begin by defining the limiting probabilities (determined by τ)

$$b_\tau(i,j) = \mathrm{pr}(R_i \text{ is left of } R_j \text{ in } D),$$
$$1 \le i,\ j \le N.$$

It follows that when τ is the replacement rule the asymptotic mean search time is

$$E_\tau(S;p) = \sum_{j=1}^{N} p_j p\{1 + \sum_{i \neq j} b_\tau(i,j)\} + \sum_{j=1}^{N} p_j(1-p)[1 + \sum_{i \neq j} \{1-b_\tau(i,j)\}]$$

$$= N - p(N-1) + (2p-1) \sum_{j=1}^{N} p_j \sum_{i \neq j} b_\tau(i,j). \qquad (2.1)$$

When $p = \frac{1}{2}$, $E_\tau(S;\frac{1}{2}) = \frac{1}{2}(N+1)$ does not depend on τ indicating that when a search for any record is equally likely to begin from either end of D the ordering of records is immaterial i.e. no heuristic is necessary, since the mean search time cannot be reduced under these circumstances.

To prove (ii) let τ be an optimal heuristic for the corresponding sequential list of N records. Then $\tau = \{\tau_1, \tau_2, \ldots, \tau_N\}$ is a set of permutations such that τ_j is applied to that list whenever the requested record is found in position j, and τ can be extended for D by defining $\tau_j^L = \tau_j = \tau_j^R$.

Since τ is an optimal heuristic for the corresponding sequential list it follows that for any other heuristic $\rho \neq \tau$ operating on the sequential list

$$\lim_{p\to 1} E_\tau(S;p) = 1 + \sum_{j=1}^{N} p_j \sum_{i\neq j} b_\tau(i,j)$$

$$\leq 1 + \sum_{j=1}^{N} p_j \sum_{i\neq j} b_\rho(i,j) = \lim_{p\to 1} E_\rho(S;p). \qquad (2.2)$$

It follows from (2.2) and (2.1) that when τ,ρ are extended for use on D

$$E_\tau(S;p) \leq E_\rho(S;p), \quad \tfrac{1}{2} \leq p \leq 1. \qquad (2.3)$$

To complete the proof of (ii) assume that there exists a heuristic σ such that for at least one $p_0 \in (\frac{1}{2},1]$

$$E_\sigma(S;p_0) < E_\tau(S;p_0) \quad \text{and} \quad E_\sigma(S;p) = E_\tau(S;p), \quad p \neq p_0.$$

Then it follows from (2.1) that

$$\sum_{j=1}^{N} p_j \sum_{i\neq j} b_\sigma(i,j) < \sum_{j=1}^{N} p_j \sum_{i\neq j} b_\tau(i,j). \qquad (2.4)$$

Clearly σ does not consist solely of a set of permutations $\{\sigma_1,\ldots,\sigma_N\}$ since, by (2.2) with $\rho = \sigma$ we obtain the contradiction $E_\sigma(S;p_0) \geq E_\tau(S;p_0)$. Therefore σ consists, at most, of the permutations $\{\sigma_1^L,\sigma_1^R,\ldots,\sigma_N^L,\sigma_N^R\}$. Now, define a new replacement rule, ω, for the corresponding sequential list. The heuristic ω is given by $\omega = \{\omega_j = \sigma_j^L, \omega_j' = \sigma_j^R\}$ where ω_j (ω_j') is applied to the sequential list whenever the requested record is found in position j and the result of a corresponding independent Bernoulli trial is $L(R)$; with respect to the

Bernoulli trial we specify that $pr(L) = p_0$ and $pr(R) = 1 - p_0$. Probabilistically, ω is equivalent to σ and therefore $b_\omega(i,j) = b_\sigma(i,j)$. Hence

$$1 + \sum_{j=1}^{N} p_j \sum_{i \neq j} b_\omega(i,j) = 1 + \sum_{j=1}^{N} p_j \sum_{i \neq j} b_\sigma(i,j)$$

$$< 1 + \sum_{j=1}^{N} p_j \sum_{i \neq j} b_\tau(i,j)$$

contradicting (2.2), and (ii) follows at once.

It follows from the lemma that in searching for optimal heuristics for D it suffices to find an optimal heuristic for the corresponding sequential list. Rivest [7] has deduced a partial characterization of an optimal replacement rule, if such a heuristic exists, and has exhibited empirical evidence which suggests that the transposition heuristic is in fact optimal for any distribution of search probabilities.

Suppose, now, that at least one of $p_{1L},\ldots,p_{NL}$ is different. We consider a modification of the move to front heuristic called "move to end" (MTE) which operates in the following fashion; let $D \equiv (R_1,R_2,\ldots,R_N)$. If R_i is requested and the associated search began at the left end of D place R_i at the left end of D and shift $R_1,R_2,\ldots,R_{i-1}$ one position to the right; then D becomes

$D' \equiv (R_i, R_1, \ldots, R_{i-1}, R_{i+1}, \ldots, R_N)$. Conversely, if R_i is requested and the associated search began at the right end of D place R_i at the right end of D and shift $R_{i+1}, \ldots, R_N$ one position to the left; then D becomes $D'' = (R_1, \ldots, R_{i-1}, R_{i+1}, \ldots, R_N, R_i)$. The mean search time for MTE is easily calculated. Let $b(i,j)$ denote the asymptotic probability that R_i is located to the left of R_j in D under MTE. Then

Lemma 3. $b(i,j) = p_i p_{iL}/(p_i + p_j) + p_j p_{jR}/(p_i + p_j)$.

Proof: At any time R_i is located to the left of R_j if one of the following mutually exclusive cases obtains:

(1) There exists a unique k_1 such that the preceding k_1 requests consist of a request for R_i, which was found by searching D from the left end, followed by $(k_1 - 1)$ requests for records other than R_i or R_j; the probability in this case is

$$p_1 = p_i p_{iL} \sum_{k_1=1}^{\infty} (1 - p_i - p_j)^{k_1 - 1} = p_i p_{iL}/(p_i + p_j).$$

(2) There exists a unique k_2 such that the preceding k_2 requests consist of a request for R_j, which was found by searching D from the right end, followed by $(k_2 - 1)$ requests for records other than R_i or R_j; the prob-

ability in this case is

$$p_2 = p_j p_{jR}/(p_i + p_j).$$

The result follows at once.

Using Lemma 3 we can derive an expression for $E_{MTE}(S)$, the expected search time under the MTE heuristic and show that $E_{MTE}(S)$ never exceeds twice the expected search time when D is optimally ordered.

<u>Theorem</u>. If $E^*(S)$ is the minimum search time when D is optimally arranged, from left to right, in decreasing order of $p_i^* = p_i(p_{iL} - \frac{1}{2})$, $1 \le i \le N$, then

$$E_{MTE}(S) < 2E^*(S).$$

<u>Proof</u>: Denote the products $p_i p_{iL}$ and $p_j p_{jR}$ by $\hat{p}_{iL}$, $\hat{p}_{jR}$ respectively, $1 \le i, j \le N$. Then

$$\begin{aligned} E_{MTE}(S) &= \sum_{j=1}^{N} \hat{p}_{jL}\{1 + \sum_{i \ne j} b(i,j)\} + \sum_{j=1}^{N} \hat{p}_{jR}\{1 + \sum_{i \ne j} b(j,i)\} \\ &= \sum_{j=1}^{N} \hat{p}_{jL} + \sum_{j=1}^{N} \hat{p}_{jR} + \sum_{j=1}^{N} \hat{p}_{jL} \sum_{i \ne j} b(i,j) + \sum_{j=1}^{N} \hat{p}_{jR} \sum_{i \ne j} b(j,i) \\ &= 1 + \sum_{j=1}^{N} \sum_{i \ne j} (2\hat{p}_{jR}\hat{p}_{jL} + \hat{p}_{jL}\hat{p}_{iL} + \hat{p}_{jR}\hat{p}_{iR})/(p_i + p_j) \\ &= 1 + 2\sum_{j=1}^{N} \{\hat{p}_{jL} \sum_{i<j} (\hat{p}_{jR} + \hat{p}_{iL})/(p_i + p_j) \\ &\quad + \hat{p}_{jR} \sum_{i>j} (\hat{p}_{iR} + \hat{p}_{jL})/(p_i + p_j)\}. \qquad (1) \end{aligned}$$

To prove the result we optimally arrange D and then relabel the records so that R_i is the ith record from the left under this ordering. Then from (1.1) it follows that the optimal expected search time is given by

$$E^*(S) = \sum_{j=1}^{N} p_j p_{jL} j + \sum_{j=1}^{N} p_j p_{jR}(N - j + 1)$$

$$= 1 + \sum_{j=1}^{N} \hat{p}_{jL}(j - 1) + \sum_{j=1}^{N} \hat{p}_{jR}(N - j). \qquad (2)$$

Since $\hat{p}_{iL} + \hat{p}_{jR} \le p_i + p_j$, $\hat{p}_{iR} + \hat{p}_{jL} \le p_i + p_j$ it follows from (1), (2) that

$$E^*(S) \le E_{MTE}(S)$$

$$\le 1 + 2\sum_{j=1}^{N} \hat{p}_{jL}(j - 1) + 2\sum_{j=1}^{N} \hat{p}_{jR}(N - j)$$

$$< 2E^*(S).$$

i.e. on average MTE never requires more than twice the search time associated with the optimal ordering of D.

In summary, then, if D is equally likely to be searched from either end for any particular record there is no heuristic which will reduce the mean search time below the value $\frac{1}{2}(N+1)$. If, however, a search is more likely to begin at one particular end of D with a constant probability for every item requested, then in order to reduce the mean search time it suffices to regard D as a sequential list and to apply known results concerning heuristics for maintaining sequential lists in approximately optimal order. Finally, if the probabilities for search direction are not constant the move to end heuristic represents one replacement rule which has the potential to reduce mean search time considerably by dynamically reorganizing the doubly linked list when the request and search direction probabilities are unknown.

References

[1] Bittner, J.R., Heuristics that dynamically organize data. SIAM J. Comput. 8 (1979), 82-110.

[2] Burville, P.J. and Kingman, J.F.C., On a model for storage and search. J. Appl. Prob. 10 (1973), 697-701.

[3] CODASYL. CODASYL Data Base Task Group Report. ACM, New York, 1971.

[4] Hendricks, W.J., An account of self-organizing systems. SIAM J. Comput. 5 (1976), 715-725.

[5] Knuth, D.E., The Art of Computer Programming, Vol. I, Fundamental Algorithms. Addison-Wesley, Reading, Mass., 1973.

[6] Knuth, D.E., The Art of Computer Programming, Vol. III, Sorting and Searching. Addison-Wesley, Reading, Mass., 1973.

[7] Rivest, R., On self-organizing sequential search heuristics. Comm. A.C.M. 19 (1976), 63-67.

[8] Tremblay, J.P. and Sorenson, P.G., An Introduction to Data Structures with Applications. McGraw-Hill, New York, 1976.